Text [Sip12] Continuing with regular expressions
Section 1.3 pages 66 - 76
Homework: Homework 3 will be released soon.

In earlier classes, we have studied:
- regular expressions
- closure properties for regularity
  - set-theoretic operators: \( \cup, \cap, - \)
    - preserve regularity
  - language operators: \( \epsilon, *, \)
    - also preserve regularity

Now, we show that \( \text{RE} \equiv \text{FA} \) 
(“regular expressions” are computationally equivalent to FAs)

Thus, when this is true, the notion of regularity, originally defined via “acceptance by FAs”, is equivalent to the “denotation by re’s”, that is,

a language \( L \) is regular \( \iff \) \( L = L(M) \) for some FA \( M \)

\( \iff \) \( L = L(r) \) for some re \( r \).

We prove 1 by arguing:

1. \( \forall \text{re } r \exists \text{FA } M \text{ such that } L(M) = L(r) \) — (2)
2. and \( \forall \text{FA } M \exists \text{re } r \text{ such that } L(r) = L(M) \) — (3)
Proving \( \exists M \) such that \( L(M) = L(r) \).

(1) \( \forall r \in FA \). 

Proof idea (details in [Sip12]): 
by induction on the structure of the given \( r \). 

Basis: When \( r = \phi \) (so \( L(r) = \emptyset \)) 

- A desired FA \( M \) is: 
  \[ \text{start} \rightarrow \emptyset \]
  \[ L(M) = \emptyset \]

When \( r = \varepsilon \) (so \( L(r) = \{\varepsilon\} \)) 

- A desired FA \( M \) is:
  \[ \text{start} \rightarrow \emptyset \]
  \[ L(M) = \{\varepsilon\} \]

When \( r = a \in \Sigma \) (so \( L(r) = \{a\} \)) 

- A desired FA \( M \) is:
  \[ \text{start} \rightarrow a \rightarrow \emptyset \]
  \[ L(M) = \{a\} \]

Induction: \( r \) is of the following possible forms 

- \( r_1 + r_2 \), \( r_1 \cdot r_2 \), \( r_1^* \) 

for some \( r_1, r_2 \in FA \) such that

\[ L(r_1) = L(M_1) \text{ and } L(r_2) = L(M_2) \]

for some FA \( M_1 \) and \( M_2 \).

We show the case when \( r = r_1 + r_2 \), as

you can see that the proof idea is identical.
to the proof of closure property of the set-theoretic operation \( \cup \) (done in earlier class).

Case when \( r = r_1 + r_2 \), with
\[
\begin{align*}
L(r_1) &= L(M_1) \\
L(r_2) &= L(M_2)
\end{align*}
\]

A desired FA \( M \) with \( L(M) = L(r) = L(r_1 + r_2) = L(r_1) \cup L(r_2) = L(M_1) \cup L(M_2) \).

Often cases of \( r \): \( r_1 \circ r_2 \), \( r_1^* \) are similar to the proofs of closure properties of language operators \( \circ \) and \( * \) (done in earlier class).
Proving (3) \( \forall FA, M, \exists r \) such that \( L(r) = L(M) \)

([Sip12] Lemma 1.60 statement)

The proof in [Sip12] requires the introduction of "generalized nondeterministic finite automata". We develop a simplified version (essentially, similar to [Sip12] page 72, Figure 1.63).

Proof idea: Since we have \( \text{DFA} \cup \text{NFA} \) \( \rightarrow \) FA,

we may assume, without loss of generality, that the given FA/M is a DFA:

\[(Q, \Sigma, \delta, 91, F)\]

with \(Q = 91, 92, \ldots, 9n3\) (so 91 is the start state)

Our goal is to show that

\[L(M) = L(r)\] for some (properly chosen) \(r\) \(\in\) \(R\).

Certainly, the choice of such \(r\) depends on the "shadow" of the DFA \(M\), but how?
Key idea 1:
For states $q_i, q_j \in \mathcal{Q}$ ($i, j \in \{1, 2, \ldots, n^3\}$)

Denote by $L_{i,j}$ the language/set of all strings $x \in \Sigma^*$ such that $M$ transits from $q_i$ to $q_j$ via the path labeled by $x$.

With this denotation,

$$L(M) = \bigcup_{q_j \in \mathcal{F}} L_{i,j}.$$ 

To see that $L(M) = L(r)$ for some $r$, it suffices to show that

$$L_{i,j} = L(r_{i,j})$$

for some $r_{i,j}$.

(hence $L(M) = \bigcup_{q_j \in \mathcal{F}} L_{i,j} = L(\sum_{q_j \in \mathcal{F}} r_{i,j})$)

and so a desired $r$ is

$$r = \sum_{q_j \in \mathcal{F}} r_{i,j}$$

Still, for $i, j \in \{1, 2, \ldots, n^3\}$, how to find $r_{i,j}$ such that $L(r_{i,j}) = L_{i,j}$?
Key idea 2: For $i, j \in \{1, 2, \ldots, n\}$

For each $x \in L_{i,j}$ (causing $M$ to transit from $q_i$ to $q_j$), $M$ may enter and leave "intermediate states".

We place a "constraint" on those intermediate states so that we can construct a desired $r_{i,j}$ such that $L(r_{i,j}) = L_{i,j}$.

Formally, for all integers $i, j \in \{1, 2, \ldots, n\}$

and all integers $k = 0, 1, 2, \ldots, n$,

denote by $L_{i,j}^k = \{ x \in L_{i,j} | M \text{ transits from } q_i \text{ to } q_j \text{ on } x$ and all intermediate states in the transition ("in" and "out") must have state-numbers $\leq k \}$

Strings $r$,

$abc \in L_{i,j}$

$bbbab \notin L_{i,j}$
Now, we are viewing lots of languages for all combinations of $i,j \leq \{1,2,...,n\}$ and $k \leq \{0,1,...,n\}$.

We'll view them inductively (on $k$) and show that $L_{ij}^k = L_r(r_{ij})$ for some properly chosen $r_{ij}$.

**Basis:** $k = 0$.

For all $i,j \leq \{1,2,...,n\}$, what is $L_{ij}^0 = L_r(r_{ij})$?

Note that the constraint "all intermediate state numbers $\leq 0$" is equivalent to "no intermediate states at all".

So

$$L_{ij}^0 = \begin{cases} +a & \text{ if } i \neq j \\ +a + \varepsilon & \text{ if } i = j \end{cases}$$

where $a$ is a constant, and $\varepsilon$ is a small positive number.

For $i = j$:

- $Q_i \xrightarrow{b} Q_j$
- $r_{ij}^0 = b + \varepsilon$

For $i \neq j$:

- $Q_i \xrightarrow{a} Q_j$
- $r_{ij}^0 = \varepsilon$

$L_{ij}^0 = \{b, d\}$ for $i \neq j$:

$$L_{ij}^0 = b + d$$

$L_{ij}^0 = \{b, c, \varepsilon\}$ for $i = j$:

$$L_{ij}^0 = b + c + \varepsilon$$
Induction: Assume that, for all $i, j \leq 1, 2, \ldots, n^3$ and with $k \leq 0, 1, \ldots, n-13$, we have:

$$L_{i,j}^k = L(R_{i,j}^k)$$

for some $res \, R_{i,j}^k$.

How can we find, for all $i, j \leq 1, 2, \ldots, n^3$, $res \, R_{i,j}^{k+1}$ such that

$$L_{i,j}^{k+1} = L(R_{i,j}^{k+1})$$

Key idea 3: Consider a typical $x \in L_{i,j}^{k+1}$ and during M's transition from $q_i$ to $q_j$ on $x$, the two cases on the intermediate-state numbers:

Case 1: all intermediate-state numbers $\leq k$ ($\leq k+1$).

So $x \in L_{i,j}^k$ ($= L(R_{i,j}^k)$)

by induction hypothesis.

Case 2: does have at least one intermediate-state number $k+1$.

Say,

\[
\begin{array}{c}
q_i \\
\downarrow \\
\vdots \\
q_{k+1} \\
\downarrow \\
\vdots \\
q_{k+1} \\
\downarrow \\
\vdots \\
q_{k+1} \\
\downarrow \\
q_j
\end{array}
\]
Combining Cases 1 and 2, we can see that a desired \( R_{ij} \) such that \( L(R_{ij}) = L_{ij} \) is:

\[
R_{ij} = R_{ij}^{k} + R_{ij}^{k+1} (R_{ku}^{k} R_{uk}^{k}) R_{ku}^{k+1}
\]

Case 1

Case 2

So, finally, we'll construct \( R_{ij} \) s.t. (for all \( i, j \in \{1, 2, \ldots, n\} \))

\[
R_{ij}^{n}
\]

such that \( L(R_{ij}^{n}) = L_{ij}^{n} \)

(\( k = n \))

But this is simply \( L_{ij} \) (see key idea 2, page 6)

Thus, a desired \( R_{ij} \) such that \( L(R_{ij}) = L_{ij} \)

\[
R_{ij} = R_{ij}^{n}
\]
Then, back to Key idea 1, page 5, a desired re $r$ such that $L(r) = L(M)$:

$$
\begin{align*}
R &= R_1 + \bigcup_{q_i \in F} \left( \bigcup_{q_j \in F} L(q_i, q_j) \right) \\
&\quad + \bigcup_{q_i \in F} L(r, q_i)
\end{align*}
$$

**Summary**: How do we find re $r_1$ such that $L(r) = L(M)$?

**Step 1**: Recursive / inductive construction (induction on $k$)

For $k = 0, 1, \ldots, n$,

$$
R_{i,j} \quad \text{for all } i,j \in \{1, 2, \ldots, n\}
$$

Then, desired re is

$$
R_{i,j} = R_{i,j} \quad \text{for all } i,j \in \{1, 2, \ldots, n\}
$$

**Step 2**: A desired re $r$ such that $L(r) = L(M)$:

$$
R = R_1 + \bigcup_{q_i \in F} \left( \sum_{j} \sum_{q_j \in F} R_{i,j} \right)
$$

Read [Sip12] Lemma 1.60 and its proof and example pages 69 - 76.